

## ESCUELA DE TECNOLOGÍAS DE LA INFORMACIÓN

**Tesis previa a la obtención de título de  
Ingeniero en Informática y Multimedia**

**AUTOR:** Wilson Eduardo Armijos Cangó

**TUTOR:** Ing. Luis Roberto Jácome Galarza

Desarrollo de una App Móvil con Inteligencia Artificial  
para traducir de Lenguaje de señas a Lenguaje natura.

## ACUERDO DE CONFIDENCIALIDAD

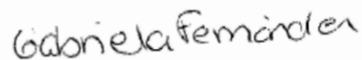
La Biblioteca de la Universidad Internacional del Ecuador se compromete a:

1. No divulgar, utilizar ni revelar a otros la **información confidencial** obtenida en el presente trabajo, ya sea intencionalmente o por falta de cuidado en su manejo, en forma personal o bien a través de sus empleados.
2. Manejar la **información confidencial** de la misma manera en que se maneja la información propia de carácter confidencial, la cual bajo ninguna circunstancia podrá estar por debajo de los estándares aceptables de debida diligencia y prudencia.



---

Wilson Armijos  
Egresado de la Carrera de Informática y Multimedia



---

Gabriela Fernández  
Gestora Cultural

# Aplicación móvil con inteligencia artificial para traducción del lenguaje de señas al lenguaje natural

Wilson E. Armijos C.

March 4, 2024

## Abstract

The fundamental objective of this project seeks to facilitate communication between people who are deaf or hard of hearing and those who do not use this language. The app uses cutting-edge technologies, such as image recognition and machine learning, to interpret and translate signs in real time. To achieve this, new technologies were implemented such as Python, OpenCV, Django, Object Detection, with their respective versions and other specific tools that will be detailed later.

Keywords: Python, Object Detection, OpenCV, Sign Language, Deep Learning, Redes Convolucionales (CNN)

## Resumen

El objetivo fundamental de este proyecto busca facilitar la comunicación entre personas sordas o con problemas de audición y aquellas que no utilizan este lenguaje. La aplicación utiliza tecnologías de vanguardia, como reconocimiento de imágenes y aprendizaje automático, para interpretar y traducir señas en tiempo real. Para lograrlo, se implementaron nuevas tecnologías como Python, OpenCV, Django, Object Detection, con sus respectivas versiones y otras herramientas específicas que se detallarán más adelante.

Palabras claves: Python, Object Detection, OpenCV, Lenguaje de Señas

## 1 Introducción

La comunicación efectiva es fundamental para el desarrollo social y emocional de todos los individuos. Sin embargo, para las personas con discapacidad auditiva, la barrera del lenguaje puede representar un desafío significativo en su interacción diaria con quienes no comprenden el lenguaje de señas. A pesar de los avances tecnológicos, la disponibilidad de herramientas eficaces para facilitar la comunicación entre personas sordas y oyentes sigue siendo limitada. Conscientes de esta problemática, el presente proyecto se centra en el desarrollo de una aplicación innovadora y accesible que aborde las necesidades específicas de la comunidad de personas sordas. Esta aplicación tiene como objetivo principal ofrecer una solución efectiva para la traducción en tiempo real del lenguaje de señas a un formato comprensible para aquellas personas que no conocen el lenguaje de señas. A través de la integración de tecnologías de vanguardia,

como reconocimiento de imágenes y aprendizaje automático, se busca facilitar la interacción y promover la inclusión de personas con discapacidad auditiva en una variedad de entornos sociales y profesionales. El presente documento detalla el diseño, la implementación y la evaluación de la aplicación propuesta, destacando su contribución potencial en el ámbito de la accesibilidad y la comunicación inclusiva. Asimismo, se abordan las consideraciones éticas y técnicas relevantes para garantizar la eficacia y la confiabilidad del sistema desarrollado. Mediante este esfuerzo, se pretende ampliar las posibilidades de interacción y promover la igualdad de oportunidades para las personas sordas en nuestra sociedad contemporánea.

## **1.1 Problema de investigación**

La OMS, mencionó al conmemorarse el Día Internacional de las Personas con capacidades especiales que más de 1000 millones de personas, que equivale a un 15% de la población mundial, padece un tipo de discapacidad y forman parte de los grupos más vulnerables del mundo. En el Ecuador, durante los años 2009 y 2013 la Misión Solidaria Manuela Espejo, realizó una investigación exhaustiva con el fin de poder determinar el número de personas que tengan un tipo de discapacidad, en esta investigación llevada a cabo se pudo constatar que en el país existen alrededor de 294.304 personas con discapacidad. La comunicación es importante ya que es el medio por el cual expresamos nuestras ideas, por otro lado, existen varios factores que impiden a las personas desarrollar su comunicación (problemas cerebrales, daños en las cuerdas vocales, etc.) lo cual dificulta la comunicación con terceras personas. Sin embargo, estamos viviendo una era donde la tecnología avanza a grandes pasos, lo cual ha permitido ayudar a estas personas a romper toda clase de barrera existente sobre todo en el ámbito social. El desarrollo de una aplicación la cual sirva como intérprete para la comunicación con personas con problemas auditivos y comunicación, es de vital importancia ya que con la ayuda de la tecnología y la aplicación a desarrollarse será un gran avance para solventar una pequeña parte de esta problemática existente. Finalmente, todos estos inconvenientes a los cuales debe enfrentarse las personas con discapacidad anteriormente mencionada, al momento de comunicarse con la sociedad serán resueltas gracias al mundo de la tecnología y uso de todas sus herramientas.

## **2 Materiales y Métodos**

### **2.1 Metodología de desarrollo**

Para el desarrollo de este proyecto, se ha seleccionado la metodología Extreme Programming (XP). Esta metodología se considera adecuada para el desarrollo del proyecto, ya que proporciona un ciclo de vida claro y estructurado que permite al programador tener una visión clara de cómo realizar el proyecto de manera efectiva. En XP, se enfatiza la colaboración estrecha entre el equipo de desarrollo y el cliente, lo que permite una comprensión profunda de los requisitos y una comunicación constante a lo largo del proyecto. Esto garantiza que el producto final cumpla con los requisitos establecidos por el cliente. Además, XP se basa en la iteración y la entrega continua de software funcional. Esto sig-

nifica que se desarrollan incrementos pequeños y funcionales del sistema, lo que permite recibir comentarios tempranos y realizar ajustes en el camino. Esto resulta especialmente útil en proyectos de desarrollo de software, ya que los cambios y las mejoras se pueden incorporar rápidamente. Otra ventaja de XP es su enfoque en la calidad del código y las prácticas de desarrollo, como la programación en pareja y las pruebas unitarias. Estas prácticas promueven la mejora continua, la detección temprana de errores y la entrega de un software robusto y confiable. En resumen, la metodología Extreme Programming ofrece un enfoque ágil y orientado a resultados para el desarrollo del proyecto de titulación. Su enfoque en la colaboración, la entrega continua y la calidad del código proporciona una base sólida para el éxito del proyecto y la satisfacción del cliente.

## 2.2 Desarrollo

### 2.2.1 Codificación

Durante esta fase del proyecto, se utilizó diversas herramientas y tecnologías que se adapten al desarrollo de la aplicación de lenguaje de señas. Algunas de las herramientas clave a considerar son las siguientes:

- Python v3.8
- Tensorflow v2.10
- TensorFlow Object Detection API
- CUDA Toolkit v11.2
- CuDNN v8.1.0
- Protobuf v3.19.6
- Visual studio c++ tools
- Nvidia GPU (GTX 650 o superior)
- COCO Api
- Anaconda Navigator
- Django
- PWA (Progressive Web Apps)

Cabe destacar que la utilización de estas herramientas se lleva a cabo dentro de un entorno virtual de Anaconda Navigator. Esta práctica se adopta con el objetivo de evitar posibles conflictos de paquetes con otros proyectos, asegurando así un ambiente de desarrollo aislado y controlado.

### 2.2.2 Creación del Dataset

Para la creación del dataset, se utilizó videos que representen el lenguaje de señas. Mediante el uso de código implementado en Python, se capturaron los fotogramas de estos videos con el objetivo de obtener un dataset completo.

Como se muestra en la figura 1, es la cantidad de fotogramas que se ha capturado utilizando el código python implementado, cuyo resultado es un conjunto de imágenes en escala de grises que representan cada uno de los fotogramas del video. Este conjunto de imágenes puede utilizarse como un dataset para entrenar modelos de reconocimiento de lenguaje de señas u otros fines relacionados con el procesamiento de imágenes.



Figure 1: Imágenes del dataset

### 2.2.3 Etiquetado de imágenes por categorías

Para llevar a cabo el proceso de etiquetado del dataset previamente creado, se utilizó la herramienta Label Studio, es necesario descargar los archivos para su uso a través de la clonación del repositorio correspondiente.

### 2.2.4 Creación de rutas

La creación de rutas para gestionar archivos es un aspecto crucial en el desarrollo de un proyecto, ya que facilita el acceso a todos los archivos necesarios de manera eficiente. Como se puede apreciar en la figura 2, se ha definido rutas específicas para cada archivo, y estas rutas se almacenan en variables o paths. Esta práctica permite acceder de manera sencilla a los archivos sin la necesidad de escribir manualmente toda la estructura de directorios en cada ocasión.

```
1 WORKSPACE_PATH = 'comun/templates/Tensorflow/workspace'
2 ANNOTATION_PATH = 'comun/templates/Tensorflow/workspace/annotations'
3 MODEL_PATH = 'comun/templates/Tensorflow/workspace/models'
4 IMAGE_PATH = 'comun/templates/Tensorflow/workspace/images'
5 PRETRAINED_MODEL_PATH = 'comun/templates/Tensorflow/workspace/pre-trained-models'
6 CONFIG_PATH = 'comun/templates/Tensorflow/workspace/models/my_ssd_mobnet/pipeline.config'
7 CHECKPOINT_PATH = 'comun/templates/Tensorflow/workspace/models/my_ssd_mobnet/'
8 CUSTOM_MODEL_NAME = 'comun/templates/Tensorflow/workspace/models/my_ssd_mobnet'
9 API_MODEL_PATH = 'comun/templates/Tensorflow/models'
10 SCRIPTS_PATH = 'comun/templates/Tensorflow/scripts'
```

Figure 2: Creación de rutas

### 2.2.5 Creación del archivo label map para las categorías del lenguaje de señas

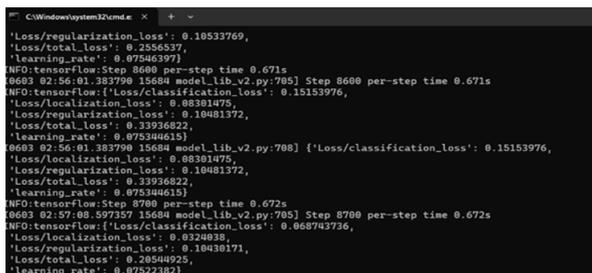
La creación de un archivo "label map" de las categorías es un proceso esencial en proyectos de aprendizaje automático que involucran la clasificación o detección de objetos en imágenes. Este archivo tiene la función de establecer una correspondencia entre las categorías que deseamos reconocer en nuestros datos y los identificadores numéricos que el modelo de aprendizaje automático utiliza internamente para representar esas categorías. En esencia, actúa como un diccionario que relaciona nombres descriptivos con valores numéricos.

### 2.2.6 Creación de archivos TF record test y train

En la creación de archivos de tipo 'test' y 'train' en formato TFRecord Estos archivos, en formato binario, son ampliamente utilizados en TensorFlow para almacenar datos de manera altamente eficiente. Son particularmente útiles para guardar conjuntos de datos extensos destinados a la capacitación de modelos de aprendizaje automático. En estos archivos TFRecord, los datos se organizan en registros, los cuales pueden contener una variedad de tipos de información, como imágenes, secuencias de texto o tensores. La ventaja clave de TFRecord radica en su eficiencia tanto en la lectura como en la escritura de datos, lo que lo convierte en una elección ideal para el procesamiento de grandes volúmenes de datos en escenarios de aprendizaje automático a gran escala.

### 2.2.7 Entrenamiento de los datos

En esta fase del proyecto, se realiza el entrenamiento utilizando el conjunto completo de datos recopilados del lenguaje de señas. En la figura 3, se presentan los resultados alcanzados en cada etapa del entrenamiento, destacando los porcentajes respectivos de aprendizaje y pérdida. Esta representación visual ofrece una visión detallada del rendimiento del modelo a lo largo del proceso de entrenamiento, proporcionando una evaluación completa de su capacidad de aprendizaje y la eficacia en la reducción de la pérdida.



```
CI\Windows\system32\cmd.exe
'Loss/regularization_loss': 0.10533769,
'Loss/total_loss': 0.2356537,
'Learning_rate': 0.07546397}
INFO:tensorflow:Step 8600 per-step time 0.671s
0603 02:56:01.283790 15684 model_lib_v2.py:705] Step 8600 per-step time 0.671s
INFO:tensorflow: {'Loss/classification_loss': 0.15153976,
'Loss/localization_loss': 0.88301475,
'Loss/regularization_loss': 0.10481372,
'Loss/total_loss': 0.33936822,
'Learning_rate': 0.075344615}
0603 02:56:01.283790 15684 model_lib_v2.py:705] {'Loss/classification_loss': 0.15153976,
'Loss/localization_loss': 0.88301475,
'Loss/regularization_loss': 0.10481372,
'Loss/total_loss': 0.33936822,
'Learning_rate': 0.075240615}
INFO:tensorflow:Step 8700 per-step time 0.672s
0603 02:57:08.597357 15684 model_lib_v2.py:705] Step 8700 per-step time 0.672s
INFO:tensorflow: {'Loss/classification_loss': 0.060793736,
'Loss/localization_loss': 0.83240398,
'Loss/regularization_loss': 0.10430171,
'Loss/total_loss': 0.205404925,
'Learning_rate': 0.075222821}
```

Figure 3: Entrenamiento de datos

## 3 Resultados

En la Tabla 1, se presentan los resultados obtenidos durante las pruebas realizadas, permitiendo así la evaluación del nivel de confianza alcanzado.

Seña	Detección	Confianza
Triste	sí	55%
Gracias	sí	51%
Levantarse	sí	51%
Te amo	sí	62%
Teléfono	sí	52%
Libro	sí	53%
Bebé	sí	74%
Animal	sí	54%
Esperar	sí	63%
Feliz	sí	78%
Bicicleta	sí	55%
Carro	sí	59%

Table 1: Resultados obtenidos



Figure 4: Señas detectadas

## 4 Discusión

Durante el desarrollo del proyecto se necesitó superar inconvenientes como la incompatibilidad de versiones de las distintas herramientas que se utilizaron. Para hacer uso del recurso de la tarjeta gráfica del ordenador, el cual permitió hacer el entrenamiento de los datos, presento dificultades ya que se debe tener una una tarjeta de video GTX 650 o superior y descargar archivos que permiten su uso. Tener en cuenta que Tensoflow v2.10 esta migrado con la v1 de Tensorflow. Para realizar el entrenamiento de la red neuronal con videos re-

quiere más capacidad de hardware. Durante el periodo de pruebas se alcanzó un nivel máximo de confianza y mínimo obtenido 78% y 51%. Una de las ventajas principales que presenta SSD (Single Shot Detector), es rápido y tiene un nivel confianza alto y una buena precisión en la detección de objetos pequeños. SSD realiza la detección objetos en un único disparo a diferencia de YOLO que divide a la imagen en múltiples celdas para realizar la detección de objetos. Las pruebas realizadas se llevaron a cabo en el CONADIS, con personas que hacen uso del lenguaje de señas.



Figure 5: Pruebas realizadas en el CONADIS

## 5 Conclusiones

El desarrollo de este proyecto representa un notable avance en la promoción de la inclusión y accesibilidad para las personas con discapacidad auditiva. A lo largo de todo el proceso, se ha evidenciado de manera contundente la importancia de crear nuevas herramientas destinadas a facilitar la comunicación, con el objetivo primordial de fomentar la igualdad de oportunidades. Durante todo el proceso de desarrollo se enfrentó a múltiples desafíos técnicos y conceptuales donde cada obstáculo superado ha sido una oportunidad para aprender y mejorar.